Dynamic Web Page Grid with JavaScript - Create Elements add Event Listeners to make page elements interactive.

```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript Code</title>
</head>
<body>
 <div class="output"></div>
 <script src="code1.js"></script>
</body>
</html>
```

```
const output = document.querySelector('.output');
console.log(output);
```

Document.querySelector()

https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector

The Document method querySelector() returns the first Element within the document that matches the specified selector, or group of selectors. If no matches are found, null is returned.

Document.createElement()

https://developer.mozilla.org/en-US/docs/Web/API/Document/createElement

In an HTML document, the document.createElement() method creates the HTML element specified by tagName,

```
const output = document.querySelector('.output');
output.textContent = 'Hello World';
console.log(output.textContent);
const div1 = document.createElement('div');
div1.textContent = 'Text 1 Hello';
console.log(div1.textContent);
const val1 = output.append(div1);
console.log(val1);
output.prepend(div1);
const val2 = output.appendChild(div1);
console.log(val2);
val2.textContent = 'Val2';
console.log(div1.textContent);
```

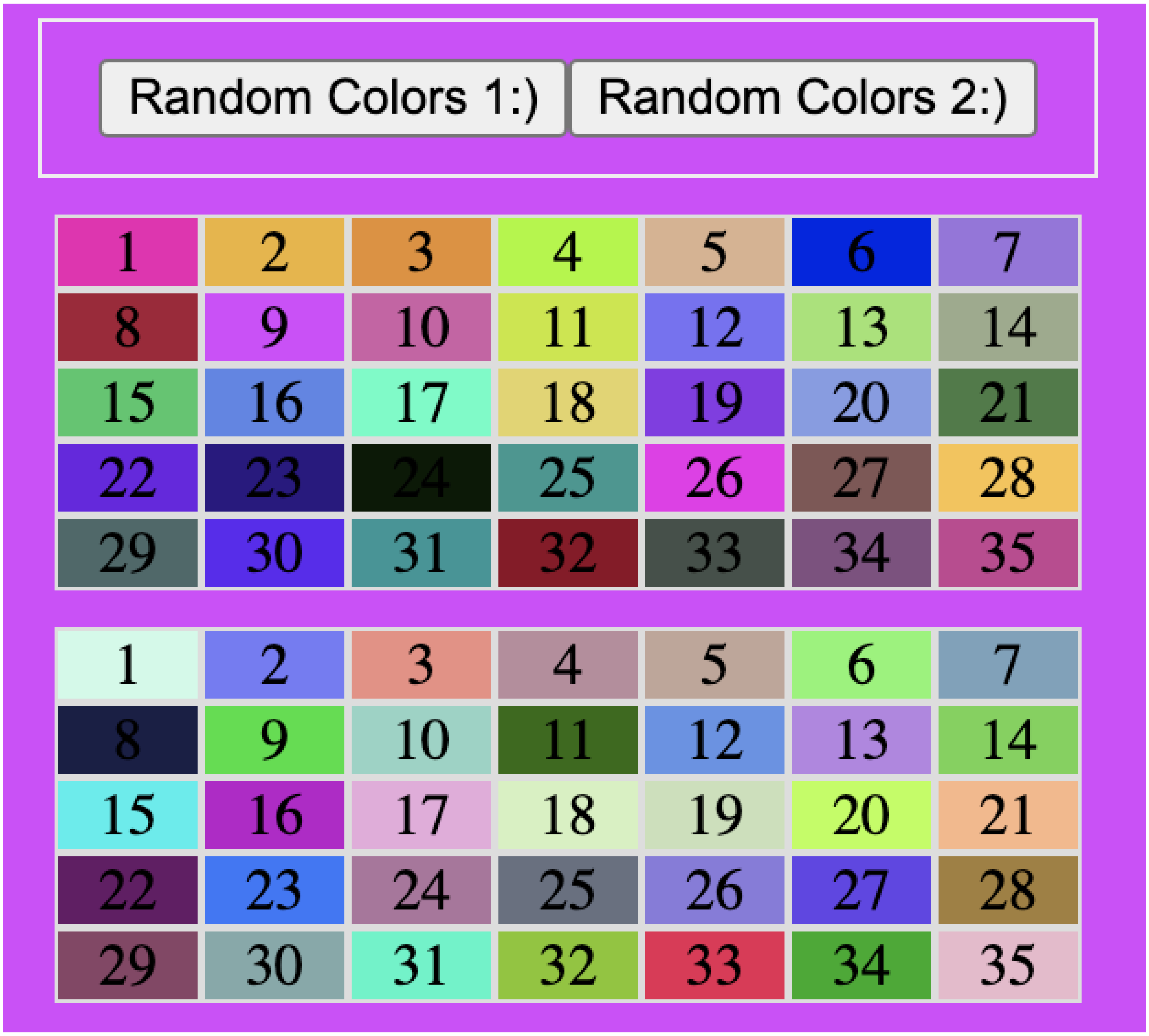
EventTarget.addEventListener()

The addEventListener() method of the EventTarget interface sets up a function that will be called whenever the specified event is delivered to the target.

https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener

```
const output = document.querySelector('.output');
const btn1 = document.createElement('button');
output.append(btn1);
btn1.textContent = 'click me 1';
const btn2 = document.createElement('button');
output.append(btn2);
btn2.textContent = 'click me 2' ;
/*
```

```
output.addEventListener('click',(e)=>{
  console.log(e.target);
  console.log(output);
},{once:true});
*/
btn1.addEventListener('click',updater);
btn2.addEventListener('click',updater);
function updater(e){
  console.log(e.target);
  //e.target.disabled = true;
  const ele = e.target;
  if(ele.style.backgroundColor == 'red'){
      ele.style.backgroundColor = 'blue';
      e.target.removeEventListener('click',updater);
  }else{
      ele.style.backgroundColor = 'red';
  }
}
```



1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35

JavaScript Dynamic and Interactive Grid Elements

```
<!DOCTYPE html>

<html>

<head>

 <title>JavaScript Code</title>

 <style>

   .box{

     border:1px solid #ddd;

     text-align:center;
```

```
    cursor:grab;
  }
  .output{
    display:grid;
    width:80vw;
    margin:auto;
  }
 </style>
</head>
<body>
 <div class="output"></div>
 <script src="code1.js"></script>
</body>
</html>
```

```
const output = document.querySelector('.output');
const grid = {rows:5,cols:7};
const total = grid.rows * grid.cols;
const output1 = document.createElement('div');

const output2 =  document.createElement('div');
output2.style.margin = '10px auto';
output2.style.border = '1px solid #eee';
output2.style.maxWidth = '80%';
output2.style.textAlign = 'center';
output2.style.padding = '10px';

const btn1 = document.createElement('button');
```

```
btn1.textContent = 'Random Colors 1:)';
const btn2 = document.createElement('button');
btn2.textContent = 'Random Colors 2:)';
const holder =[];
btn1.addEventListener('click',(e)=>{
   holder.forEach((el)=>{
       el.style.backgroundColor = ranBack();
   })
})
btn2.addEventListener('click',(e)=>{
   const eles = document.querySelectorAll('.box');
   eles.forEach((el)=>{
       el.style.backgroundColor = ranBack();
   })
})

document.body.prepend(output1);
document.body.prepend(output2);
output2.append(btn1);
output2.append(btn2);
output1.classList.add('output');
output1.style.marginBottom = '10px';
createGrid(total,output,grid.cols);
createGrid(total,output1,grid.cols);

function  ranBack(){
   return '#'+(Math.random().toString(16).substr(-6));
}
```

```
function createGrid(tot,parent,cols){
   for(let i=0;i<tot;i++){
       const ele = maker(i,parent);
       holder.push(ele);
       ele.addEventListener('click',updateBack);
   }

parent.style.setProperty(`grid-template-columns`,`repeat(${cols}
,1fr)`)
}

function updateBack(e){
   const ele = e.target;
   console.log(ele.style.backgroundColor);
   document.body.style.backgroundColor =
ele.style.backgroundColor;
}


function maker(i,parent){
   const ele = document.createElement('div');
   ele.textContent = `${i+1}`;
   ele.classList.add('box');
   ele.style.backgroundColor = ranBack();
   return parent.appendChild(ele);
}
```